Automatic Text Summarization Using Lexical Clustering

Kono Kim, Youngjoong Ko, Jungyun Seo

Department of Computer Science
Sogang University
Shinsu-dong, Mapo-gu, Seoul Korea, 121-742
{kono, kyj}@nlprep.sogang.ac.kr, seojy@ccs.sogang.ac.kr

Keywords: text summarization, lexical clustering, k-Means algorithm

Contact Author: Kono Kim (Phone: 82-2-706-8954, E-mail: kono@nlprep.sogang.ac.kr)

Under consideration for other conferences (specify)? No.

Abstract

The goal of automatic text summarization is to reduce the size of a document while preserving its content. We investigate a summarization method which uses not only statistical features but also the contextual meaning of documents by using lexical clustering. We present a new method to compute lexical cluster in a text without high cost knowledge resources; the WordNet thesaurus. Summarization proceeds in five steps: the words of a document are vectorized, lexical clusters are constructed, topical clusters are identified, representative words of a document are selected, and a summary is produced using query. Compared with other methods, we achieved better performance at 30%, 10% and fixed 4 sentences summary experiments.

Automatic Text Summarization Using Lexical Clustering

Abstract

The goal of automatic text summarization is to reduce the size of a document while preserving its content. We investigate a summarization method which uses not only statistical features but also the contextual meaning of documents by using lexical clustering. We present a new method to compute lexical cluster in a text without high cost knowledge resources: the WordNet thesaurus. Summarization proceeds in five steps: the words of a document are vectorized, lexical clusters are constructed, topical clusters are identified, representative words of a document are selected, and a summary is produced using query. Compared with other methods, ours achieved better performance at 30%, 10% and fixed 4 sentences summary experiments.

1 Introduction

Text summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs (Mani, 2001). Several summarization methods have been investigated. These methods can be divided into two types of approach. One approach is based on linguistic analysis such as semantic distances between words and discourse structure of documents. The other approach is based on statistical analysis using title, term frequency, location of sentence, length of sentence, and clue words.

A linguistic approach for automatic text summarization tries to understand the contextual meaning of document itself. Bazilay and Elhadad (1999) constructed lexical chain by calculating semantic distance between words using the WordNet thesaurus (Miller, 1990). Strong lexical chains are selected. The sentences related to strong chains are chosen as a summary. Marcu (1996) constructed discourse structure of a document. Salience of information can be determined based on the discourse structure. These methods can produce a summary with high quality. However, linguistic resources such as the WordNet thesaurus or high performance parser are required. These limitations lead to high cost for extending linguistic resources and slow execution time for summarization.

A statistical approach is to extract useful features from training corpus. Word frequency, title, location of sentence, length of sentence, and clue words are well known as good statistical features. Sentences or passages for a summary are selected by scores which are calculated by these statistics (Edmundson, 1999). Also Kupiec, et al. (1995) applied machine learning method to learn statistical parameters. Statistical based methods are fast and its implementations are easy. The most severe limitation of these statistical based methods is their dependence on the text genre (Barzilay and Elhadad, 1999).

Our method uses linguistic and statistical method together. To overcome the limitation of linguistic and statistical method, linguistic knowledge is constructed automatically using co-occurrence information. Then statistical methods are utilized with this linguistic knowledge. So the limitation of the both methods can be avoided. Moreover our method can be used with title less documents.

The rest of this paper is organized as follows. Section 2 explains the proposed method in overall manner. Section 3 explains lexical vector space model that is used for vectorization of words. In section 4, we explain lexical clustering which groups similar words together. The purpose of lexical clustering is to identify topical clusters and to select representative words. Then, section 5 explains a summary generation process through title query and representative words query. Section 6 is devoted to experiment results and evaluation. The last section 7 makes a conclusion and describes future works.

2 Overall Architecture

In order to produce a summary with high quality, topic of a document should be recognized and represented. A few words from a document, which have strong relations with the topic of document, are a good representation form of a document topic. We call it representative words. The problem is how to recognize the representative words. To solve this problem, we proceed in four steps: the words of a document are vectorized, lexical clusters are constructed, topical clusters are identified and representative words of a document are selected.

First step for finding representative words is to represent words as a vector. In vector space, similar words should have similar vector value. We constructed lexical vector space with co-occurrence information. The lexical vector space contained about 60,000 words of proper nouns and general nouns. Each lexical vector is represented by the co-occurrence value with the other words. The articles in the newspapers for 2 years are used in order to calculate the co-occurrence value between the words. If any two words in the lexical vector space have a similar co-occurrence pattern, the meanings of these two words is likely to be the same. Accordingly, the similarity of meaning between two words increases in keeping with the inner product of two vectors which represent the words.

The next step is a lexical clustering. The words written in the documents are converted to the lexical vectors. The converted lexical vectors are clustered by k-Means algorithm. In third step, we identify topical clusters. To identify them, we developed scoring measure for cluster. The score of cluster increases with the normalized sum of term frequency within the cluster. The topical clusters can be identified by the cluster score.

Fourth step is to select so called representative words which regards as having strong relations with the topic of document. Representative words are selected from topical clusters by term frequency.

After finding representative words, we use it as a query. Candidate sentences for a summary are extracted through representative words query and title query respectively. A summary is extracted from the candidate sentences by the following criteria.

1. The sentences being extracted in common by each query.
2. The sentences located in leading position have priority as a summary for the same time selected sentences.

The document without title is possible to be summarized by title query only. The following figure 1 shows the architecture of the proposed method.

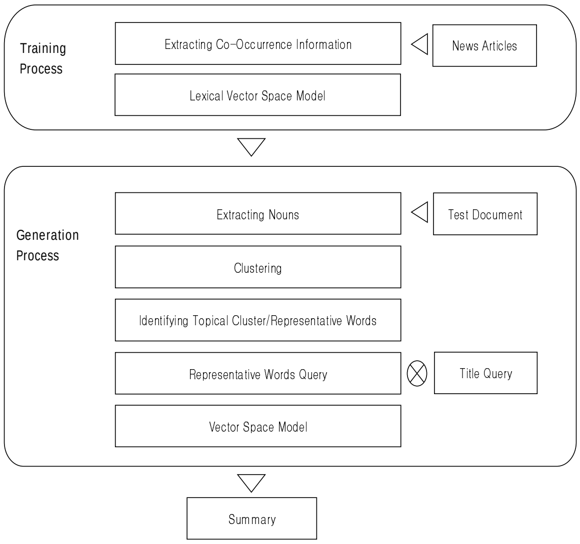

Figure 1. Automatic Text Summarization Using Lexical Clustering

3 Lexical Vector Space Model

We use lexical vector space model to represent words as a vector (Lund and Burgess, 1996). Each words in lexical vector space can be calculated the meaning similarity between words by the same method used in document vector space model. Document vector space model (Salton, 1989), which is one of well known information retrieval models, is used for representation of documents as a vector. The document similarity can be calculated by inner product and cosine.

Each word in lexical vector space is represented as the co-occurrence value with other words as seeing figure 2.

	market	company	plan	game	space
shop	*	*	*	*	
price	*	*	*		
virtual		*		*	*

* = calculated co-occurrence value

Figure 2. An Example of Lexical Vector Space When Document Words = {shop, price, virtual}, The Elements of Vector = {market, company, plan, game, space}

If any two words have a same co-occurrence pattern, the meaning of these words is similar. Accordingly, the meaning similarity between two words increases in keeping with the inner product and cosine of two vectors which represent the words.

In order to make lexical vector space, a co-occurrence value of words pair is required. For this information, the articles of Chosun Ilbo newspaper in 1996-1997 (approximately 16,600,000 words, 1,538,320 sentences) are used. The articles were tagged using morphological analyzer. A proper noun and a general noun are targeted only for calculation of each word's co-occurrence value. In the calculation of co-occurrence value, low frequency words having less than 3 in term frequency is exempted. At this time one sentence is used for the size of sliding window in order to measure co-occurrence frequency. The word pairs used in the same window is calculated to happen once. And among word pairs resulted as like this, unrelated word pairs are eliminated according to the value of mutual information as the followings formula 1

$$I(x, y) = \log \frac{\Pr(x \wedge y)}{\Pr(y) \times \Pr(y)} \qquad (1)$$

where Pr(x) is the probability of word x occurring, Pr(y) is the probability of word y occurring, and Pr(x^y) is the probability of word x and y occurring at the same time. Through this progress, a total of 2,429,342 of noun pairs were created. Lexical vector space is expressed as the co-occurrence value of two words as the formula 2.

$$\cos(X, Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \qquad (2)$$

When 60,000 words are vectorized with the other 60,000 words, bulky metrics of 60,000 by 60,000 matrixes makes a trouble at the running speed and memory space. As a result, it will be difficult to be applied at the practical application. According to the lexical vector representation research (Lund and Burgess, 1996), a similar performance achieved with 140,000 dimensional vectors and 200 dimensional vectors. This research means that it is not necessary to use the co-occurrence value of all words in order to vectorize the words.

We made dimensionality reduction experiments for deciding vector dimension with the appropriate running speed and performance. The elements for representing words were selected from Chosun Ilbo newspaper according to high term frequency. Performance results with our method were tested from 750 dimensions to 150 dimensions. The reason why we selected the elements according to frequency is to reduce data sparse problem. Word pairs with low frequency have a tendency not to have co-occurrence information. The best performance was reached at 450 dimensions as shown Figure 3.

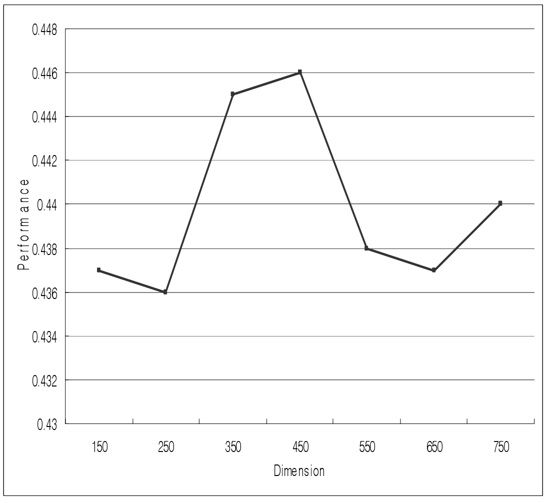

Figure 3. Performance with Various Vector Dimension

Lexical vector space is constructed during training process in advance.

4 Lexical Clustering

We vectorize words using lexical vector space model. The purpose of the modeling is to extract representative words. We can not extract representative words from lexical vector space directly. A distinction between topic related words and the others should be realized. Thus we cluster the words into several groups with similar words together. Then score is given for each cluster.

Our supposition that representative words are in topical clusters is very similar to the idea of Barzilay and Elhadad (1999) in that strong lexical chains have strong relations to express document topic. In this paper, we use lexical clusters which consist of one or more lexical chains as a model of the source text for the purpose of producing a summary. Lexical chains are the structure of similar words in a document (Moris and Hirst, 1991). Lexical clusters have different semantic categories while loosely connected each other than lexical chains. We can recognize the topic of a document by clustering words into several categories and selecting the most topic related cluster which contains topic-related words.

Nouns will be extracted from a given document for vectorization. The nouns are vectorized using lexical vector space model. These vectors are clustered using k-Means algorithm. Inner product is used for similarity measure in k-Means algorithm.

k-Means algorithm is very effective algorithm in unsupervised manner for clustering. To use k-Means algorithm, we need to set initial number of clusters. The knowledge about the number of clusters is unknown. We assumed that the number of clusters increases with the distinct number of words in the document. To determine the initial number, the *k* value decreased by a quotient of the word count divided by 20 to 65.

As a result, to set the number of clusters as the quotient of the word count divided by 50 reached good performance as shown in Figure 4.

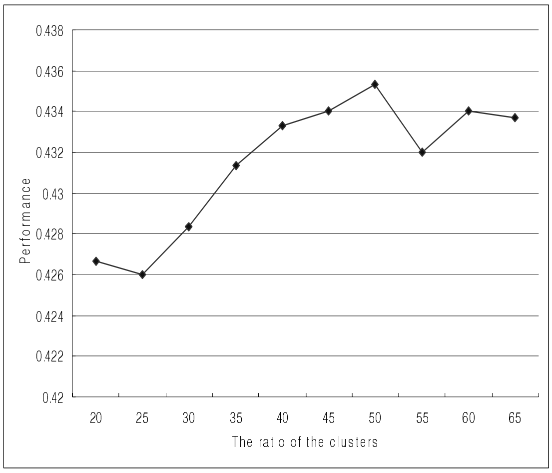

Figure 4. Performance with Various *k* Values

To determine which clusters are topical cluster, the following formula 3 is used.

$$\text{Score(Cluster)} = \frac{\sum_{i=1}^{n} tf_i}{n} \qquad (3)$$

Where *tf* is the term frequency of the word, *n* is the number of cluster members. The cluster score increases with the frequency of words and decreased with the number of cluster members. This supposes that topic related words are more frequent while only a few topic related words contribute the identification of topic clusters.

After scoring clusters, we need strategy to select topical clusters. There are several strategies such as rank-cut, percentage-cut, and score-cut. The rank-cut is strong at precision but weak at recall while the percentage-cut is vice versa (Yang, 2001). Score-cut is hard to determine the criterion.

This paper uses the combination of rank-cut and score-cut. The cluster with top score is selected by rank-cut. Then the clusters with less than 10% margin with the score of top cluster are selected as topical clusters. This strategy of cluster selection makes our method flexible at various compression rate of summarization. We can trade recall with precision by controlling the margin with the top score.

The following step selects representative words from topical clusters. The selection criteria of representative words from topical clusters are the same as selecting topical clusters from clusters. Representative words are the top frequency words within topical clusters. The words with less than 10 % margin in frequency with the top frequency word are also selected as representative words. For a summary at high compression rate, the lower margin with top score and top frequency is preferred. The lower margin results high precision and low recall. We can vary the margin at various compression rate of summary.

5 Query based Summarization

Query based summarization is one of statistical methods. This method is to make a summary by extracting relevant sentences from a document (Goldstein, 1999). The criterion for extraction is given as a query. In summarization, the probability of being included in a summary increases with the number of common words in the query and the sentence. The sentences in summary are regarded as having more information related to the topic. Usually, title is used for a query. We suggest our method as query based summarization for the following reasons.

Firstly, it is said to be important to take into account the purpose for which the produced summaries are to be used (Sparck Jones, 1998). Summaries would be more useful if summaries are produced with taking into account the purpose of the users. Query based system can easily produce user-oriented summary by adding user-defined words to a query.

Secondly, query based system can rank and score sentences according to the similarity with the query. This makes our method be flexible at various summary sizes.

Thirdly, other statistical features such as title, clue words, length of sentence, and location of the sentence can be utilized with a query. Especially, title is very good feature comparing with other features (Myaeng and Jang, 1998).

This paper uses two queries for extraction. One query consists of title and the other consists of representative words which are constructed by lexical clustering. The summary is sentences extracted in common by each query using a title and representative words. In the case of that the size of summarization is small with this selection criterion, the sentences of high position in the document are selected among the sentences picked up by any one of query. By this means, this method selects a summary which is similar to title and representative words.

The similarity measure between query and sentences is inner product. To represent sentences, only proper and general nouns are used after excepting stop words. In vectorization, Boolean weighting is used as follows.

$$S_{ik} = \begin{cases} 1 & \text{if } \mathrm{tf}_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Where *tf* is the term frequency of word *i* in sentence *k*. In general *tf.idf* representation has known to show better performance in information retrieval. However, binary representation showed better performance in summarization. The reason is that the role of words in global context is less important than the role of words in local context in summarization.

The following is a formula 5 to calculate the similarity of two vectors in document vector space model (Salton, 1989).

$$sim(i, j) = \sum_{k=1}^{n} S_{ik} S_{jk} \tag{5}$$

n is the number of nouns which is included in a document. S_{ik} is *i*-th sentence with *k*-th noun. The longer sentences are likely to be included as summary because similarity measure is inner product.

This paper is partially using statistical features such as title, location, frequency, and length of sentence. But our method is different in that title less document can be summarized with representative words query which selected by lexical clustering and can overcome the limitation of statistical method about the dependence on the text genre (Barzilay & Elhadad, 1999).

6 Experiments

In experiments, we used summarization test set of Korea Research and Development Information Center. This data is news articles. Each document has title, content, 30 % summary, 10 % summary, and manual summary. 30% and 10% summary are made by extracting sentences from content. A manual summary are made by generating sentences by human. Though 1,000 document-summary pair were reported (Kim, 1999), we used 816 document-summary pair after dropping duplicate articles and inadequate summary pairs. Statistical features of this test set are as the following table 1.

Table 1. Statistical Features of Experimental Data

Total number of documents	816
Total number of sentences	13,358
Total number of 10 % summary sentences	1,348
Total number of 30 % summary sentences	3,594
The average number of sentences per a document	16.37
The average number of sentences at 30 % summary per a document	4.40
The average number of nouns at title	6.78
The average noun number per a sentence	11.97

To make a summary at fixed length is more appropriate than making summary at certain compression rate because a summary is not related to the length of a document (Goldstein, 1999). We experimented at various compression rates (10%, 30%) and a fixed length (4 sentences).

To measure the performance of our method, F_1 measure is used. The following formula 6 is F_1 measure

$$F_1 = \frac{2(P \times R)}{P + R} \tag{6}$$

where P is precision and R is recall. When we experiment at a fixed 4 sentence summary, the F_1 measure is underestimated at recall if a document has 5 or more summary sentences. To solve this problem, we used modified F_1 measure as the following formula 7 and 8 at fixed 4 sentences summary.

$$R' = \frac{C}{\min(H, \mathrm{O})} \tag{7}$$

$$F_1' = \frac{2(P \times R')}{(P + R')} \tag{8}$$

C is the total number of correct sentence from the method, H is the total number of correct sentence, O is the total number of summary sentence from the method (In our test, O is always 4).

Our method is compared with the following 3 methods

Title Method: The score of sentences is calculated as how many words the sentence has the same word used in title. This calculation is acquired by query with title in Boolean weighted vector space model.

Location Method: It has been said that the leading few sentences of an article be important and a good summary (Wasson, 1998). Therefore, the first compression rate sentences or fixed 4 sentences are extracted as a summary by location method.

Frequency Method: The frequency of term occurrences within a text has often been used for calculating the importance of sentences (Zechner, 1996). In this method, the score of a sentence can be calculated as the sum of the score of words in the sentence. Therefore, the score of importance w_i of word i can be calculated by the traidtional *tf.idf* method (Salton, 1989), as follows.

$$w_i = tf_i \times \log \frac{N}{df_i} \tag{9}$$

where tf is the term frequency of word i in the text, N is the total number of texts, and df is the document frequency of word i in the whole set of texts.

The proposed method was measured 3 times. The performance of the proposed method is the average of 3 results. The experiment results are as shown Table 2.

Table 2. Experiment Results

Method	**30%**	**10%**	**4 Sentences**
Proposed Method	**51.1**	**51.2**	**53.6**
Title	48.6	43.3	51.6
Location	49.4	46.6	51.6
Proposed Method*	44.6	39.6	47.1
Frequency	35.9	14.8	38.4

The location method showed good performance because the leading few sentences of a news article is important and could be used as good summary.

The proposed method* used only representative words query. The proposed method* showed lower performance than that of title and location method. But the proposed method* can be very useful in the condition of summarization of title less text and non news genre.

The proposed method with two queries showed better performance at 30%, 10%, and 4 sentences all. Especially, compared with title method, our method made 2.5 % improvement at 30 % compression rate and 7.9 % improvement at 10 %. This result explains our method is stronger at high compression rate. Usually, a summarization at high compression rate requires a summary to be included only theme related sentences. Good results at 10 % summary means representative words query select topic related sentences well at high compression rate. Our method was successful at constructing representative words by lexical clustering. This result is hard to be achieved using only statistical methods. The following table 3 shows the statistical features of lexical clustering at this experiment.

Table 3. Statistical Features of Experiment Result

The average noun number per a document	195.94
The average cluster number per a document	2.98
The average topical cluster number per a document	1.38
The average representative word number per a document	1.79

7 Conclusions and Future Work

Two queries using title and representative words showed better performance than other methods. To substitute the knowledge of human, lexical vector space is constructed through collecting co-occurrence value from large corpus. This lexical vector space is successful in clustering words with similar meaning. Also, comparing with the WordNet thesaurus, lexical vector space can be easily extended and constructed. Our method exploits other good statistical features such as length of sentence in calculating similarity between query and sentence, location in selecting summary from summary candidate, frequency in selecting representative words from topical clusters. Title less and non news genre text can be summarized with our method.

We plan to research on the following problems: The number of cluster increases with the number of words. We think to set the number of clusters according to the distance of words is more appropriate. To improve the calculation of semantic distance between words, we will study other lexical vector space models.

References

R. Barzilay and M. Elhadad, Using Lexical Chains for Text Summarization, 1999. *Advances in Automatic Summarization, The MIT Press:* 111-121.

C. Burgess, K. Lund, Modeling cerebral asymmetries of semantic memory using high-dimensional semantic space, 1997. *Getting it right: The cognitive neuroscience of right hemisphere language comprehension, Hillsdale, N.J.*

H. P. Edmundson, New Methods in Automatic Extracting, 1999. *Advances in Automatic Summarization, The MIT Press:* 23-42.

J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, Summarizing Text Documents: Sentence Selection and Evaluation Metrics, 1999. *Proceedings of ACM-SIGIR' 99* : 121-128

T. H. Kim, H. R. Park, J. H. Shin, “Research on Text Understanding Model for IR/Summarization/Filtering”, 1999. *The Third Wordshop on Software Science, Seoul, Korea*

J. Kupiec, J. Pedersen, and F. Chen, A Trainable Document Summarizer, 1995. *Proceedings of ACM-SIGIR' 95* : 68-73

D. Marcu, Building Up rhetorical Structure Trees, 1996. *Proceedings of the 13th National Conference on artificial Intelligence,* Vol 2,: 1069-1074

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, 1990. *Introduction to WordNet: An on-line lexical database. International Journal of Lexicography (special issue)* 3(4):234-245

I. Mani, *Automatic Summarization,* 2001. *John Benjamins Publishing Co.*:1-22

J. Morris, G. Hirst, Lexical cohesion computed by thesaural relations as an indicator of the structure of text, 1991. *Computational Linguistics* 17(1):21-43

S. H. Myaeng, D. H. Jang, Development and Evaluation of a Statistically-Based Document Summarization System, 1998. *Advances in Automatic Summarization, The MIT Press:* 61-70

G. Salton, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, 1989. *Addison-Wesley Publishing Company*

K. Sparck Jones, Automatic sumarizing: factors and directions, 1998. *Advances in Automatic Summarization, The MIT Press:* 1-12

M. Wasson, Using leading text for news summaries: Evaluation results and implications for commercial summarization applications, 1998. *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the ACL,* pp1364-1368

Y. Yang, A Study on Thresholding Strategies for Text Categorization, 2001. *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*: 137-145,

K. Zechner, Fast generation of abstracts from general domain text corpora by extracting relevant sentences, *Proceedings of the 16th international Conference on Computational Linguistics*, pp 986-989